# Page:1

# Employees Database

### **Installation**

* npm install
* npm run dev

# **Problem**

### **Folder structure**

* public
* src
  + Components
    - EmployeesTable.jsx
    - Pagination.jsx
* package.json
* [README.md](http://README.md)

# **API Details**

Base URL : https://dbioz2ek0e.execute-api.ap-south-1.amazonaws.com/mockapi/get-employees

### \*\*Query Params\*\*

1. `page` - type = number; optional = yes; works : together with limit param; value/values can be : 1,2,3,...

2. `limit`type = number; optional = yes; works : together with page param; value/values can be : 1,2,3,...100 Example using page and limit params

`https://dbioz2ek0e.execute-api.ap-south-1.amazonaws.com/mockapi/get-employees?page=1&limit=10`

1. `filterBy` You can filter by department or gender

2. `filterValue`

If filterBy value is gender | filterValue can be male,female,others

# Page:2

If filterBy value is department | filterValue can be hr,marketing, finance, engineering, operations

Example : `https://dbioz2ek0e.execute-api.ap-south-1.amazonaws.com/mockapi/get-employees?page=1&limit=10&filterBy=department&filterValue=hr`

### \*\*Note - Make sure you use only the given components and don't create new files and folders. Changing the component name, and structures might result in giving you zero marks.\*\*

# **Description**

Create a react application that fetches the employees data , filters the data by department along with pagination. Given two components

### EmployeesTable**.jsx**

* This is the dashboard where user can see the employee data and can filter the data by department.
* user can go through different pages and fetch the relevant data.
* Functionalites to implement
* This component has different sections
  + Department dropdown with className department\_list
    - it should have the following values as options (case sensitive)
    - -Select Department-- (default option)
    - hr
    - finance
    - marketing
    - engineering
    - operations

- Table

- S.no

- Name - should have `classname = name`

- Gender - `classname = gender`

- Department - `classname = department`

- Salary - `classname = Salary`

- On initial loading of the page make a fetch request to the following API with default query params page 1 and limit 10

# Page:3

- Populate the pagination button according to the reponse

- if totalpages 10 there should be 10 button with 1,2,3...n written on it.

- The active/current page button should be in different style and should be in disabled state.

- Whenever a new page clicked request an API call and render the data.

# **DropDown functionality**

* On selecting any of the departments from the dropdown make fetch call to the given api with page,limlit,filterBy and filterValue as params
* User should be able to go through all pages of filtered data and view the results
* Make sure You are on first page whenever a new department from the list is selected.
* The active/current page button should be in different style and should be in disabled state.

# **Pagination**

* This component should receive the following props
  + totalPages
  + handlePageChange - callback function (onclicking a page send the page number back as an argument)
  + currentPage

# PROGRAM

# main.jsx

import ReactDOM from "react-dom/client";

import App from "./App.jsx";

import "./index.css";

ReactDOM.createRoot(document.getElementById("root")).render(<App />);

# Page:4

# App.jsx

import "./App.css";

import EmployeesTable from "./Components/EmployeesTable";

function App() {

return (

<div className="App">

<h2>Employees Database</h2>

{/\* Import EmployessTable.jsx \*/}

<EmployeesTable />

</div>

);

}

export default App;

# Components

# EmployeesTable.jsx

import { useEffect, useState } from "react";

import Pagination from "./Pagination";

import axios from "axios";

const EmployeesTable = () => {

const [data, setdata] = useState([]);

const [totalPages, settotalPages] = useState(1);

# Page:5

const [page, setpage] = useState(1);

const [selectdep, setselectdep] = useState([]);

console.log(totalPages);

const getdata = async () => {

try {

let res = await axios.get(

"https://dbioz2ek0e.execute-api.ap-south-1.amazonaws.com/mockapi/get-employees",

{

params: {

page: page,

limit: 10,

filterBy: "department",

filterValue: selectdep,

},

}

);

console.log(res);

setdata(res.data.data);

settotalPages(res.data.totalPages);

} catch (error) {

console.log(error);

}

};

const handlePageChange = (value) => {

setpage(value);

};

useEffect(() => {

getdata();

# Page:6

// eslint-disable-next-line react-hooks/exhaustive-deps

}, [selectdep, page]);

return (

<div>

<div>

<div>

{/\* implement Department dropdown here \*/}

<select

name=""

id=""

className="department\_list"

onChange={(e) => setselectdep(e.target.value)}

>

<option value="Select Department" hidden>

-Select Department--

</option>

<option value="hr">hr</option>

<option value="finance">finance</option>

<option value="marketing">marketing</option>

<option value="engineering">engineering</option>

<option value="operations">operations</option>

</select>

</div>

</div>

<div className="table\_container">

<table>

<thead>

<tr>

<th>S.No</th>

# Page:7

<th>Name</th>

<th>Gender</th>

<th>Department</th>

<th>Salary</th>

</tr>

</thead>

<tbody className="tbody">

{/\* map the rows here \*/}

{data.length > 0 &&

data.map((el) => (

<tr key={el.id}>

<td>{el.id}</td>

<td className="name">{el.name}</td>

<td className="gender">{el.gender}</td>

<td className="department">{el.department}</td>

<td className="Salary">{el.salary}</td>

</tr>

))}

</tbody>

</table>

</div>

{/\* import Pagination component here \*/}

<Pagination

totalPages={totalPages}

handlePageChange={handlePageChange}

currentPage={page}

/>

</div>

);

# Page:8

};

export default EmployeesTable;

# Pagination.jsx

/\* eslint-disable react/jsx-key \*/

/\* eslint-disable react/prop-types \*/

function createArrayOfSize(n) {

return new Array(n).fill(0);

}

function Pagination({ totalPages, handlePageChange, currentPage }) {

console.log(totalPages);

let pages = createArrayOfSize(totalPages).map((a, i) => {

return (

<button

data-testid="page-btn"

style={{

color: currentPage == i + 1 ? "teal" : "black",

backgroundColor: currentPage == i + 1 ? "black" : "white",

}}

disabled={currentPage == i + 1}

onClick={() => handlePageChange(i + 1)}

>

{i + 1}

</button>

);

});

return <div>{pages}</div>;

# Page:9

}

export default Pagination;

# App.css

/\* app.css \*/

.App {

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

min-height: 100vh; /\* Full height of the viewport \*/

background-color: #f5f5f5; /\* Light background color \*/

}

.department\_list {

font-family: Arial, sans-serif;

background-color: #4CAF50; /\* Green background \*/

color: white; /\* White text \*/

padding: 10px; /\* Some padding \*/

border: none; /\* Remove default border \*/

border-radius: 4px; /\* Rounded corners \*/

cursor: pointer; /\* Pointer cursor on hover \*/

margin-bottom: 20px; /\* Space below the dropdown \*/

}

# Page:10

.department\_list option {

color: black; /\* Black text for options \*/

}

.table\_container table {

width: 100%;

border-collapse: collapse;

margin: 20px 0; /\* Space around the table \*/

}

.table\_container th, .table\_container td {

border: 1px solid #ddd;

padding: 8px;

}

.table\_container th {

background-color: #4CAF50; /\* Green background for header \*/

color: white; /\* White text for header \*/

}

.table\_container tr:nth-child(even) {

background-color: #f2f2f2; /\* Light grey background for even rows \*/

}

.table\_container tr:hover {

background-color: #ddd; /\* Grey background on hover \*/

}

.table\_container th, .table\_container td {

# Page:11

text-align: left; /\* Left-align text \*/

}

.table\_container th {

padding-top: 12px;

padding-bottom: 12px;

}